AFRL-IF-RS-TR-2004-174
Final Technical Report
June 2004

MEDIUM ACCESS CONTROL FOR XG COMMUNICATIONS

BBN Technologies

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. N801

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-174 has been reviewed and is approved for publication

APPROVED: /s/
RICHARD D. HINMAN
Project Engineer

FOR THE DIRECTOR: /s/
WARREN H. DEBANY, JR.
Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2004	FINAL Aug 02 – Dec 03

4. TITLE AND SUBTITLE

MEDIUM ACCESS CONTROL FOR XG COMMUNICATIONS

5. FUNDING NUMBERS

C - F30602-02-C-0131
PE - 63760E
PR - N801
TA - XG
WU - 01

6. AUTHOR(S)

Ram Ramanathan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

BBN Technologies
10 Moulton Street
Cambridge MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

AFRL/IFGC
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRLIF-RS-TR-2004-174

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard Hinman/IFGC/(315) 330-3616 Richard.Hinman@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT ***(Maximum 200 Words)***

This report describes the results of Phase I of the architecture development in the DARPA NeXt Generation (XG) Communication program. XG is a multiphase program to define and demonstrate an architecture which will support the ability of communication and surveillance systems to sense the RF spectrum and adaptively use channels which are not in use on a non-interference basis. This report contains an overview of the project accomplishments, details can be found in a number of “sister” documents and Request for Comments (RFCs) distributions (available on the DARPA website) including: The XG Vision RFC describes the vision and goals of the XG program in general. The XG Architecture RFC presents the architecture, system components, and a high level conops for XG communications. The XG Abstract Behaviors RFC identifies key behaviors that must be implemented by an XG system. The XG Policy Language RFC describes the policy specification meta-language for implementing machine-understandable policies. The XG Policy Interface RFC describes the interface, or access methods, that can be used to pass information between an XG device and a policy reasoning engine. The XG Evaluation Platform describes the OPNET models of XG protocols.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Media Access Control, Adaptive Communication, Spectral Reuse, Policy Language	18
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

LIST OF FIGURES

1 Scope

This is a report on the *Medium Access Control for XG Communications (X-MAC)* project. The X-MAC project was funded under the DARPA NeXt Generation (XG) Communications program (Phase I), beginning in August 2002, and ending in November 2003.

This report contains an overview of the project accomplishments. We emphasize that this report is only an overview, and details can be found in a number of "sister" documents and Request for Comments (RFCs) contained in the X-MAC distributions, including:

1. The *XG Vision RFC* [XGV], which describes the vision and goals of the XG program in general and X-MAC in particular.
2. The *XG Architecture RFC* [XGAF], which presents the architecture, system components, and a high level concept of operations for XG communications.
3. The *XG Abstract Behaviors RFC* [XGAB], which identifies key behaviors that must be implemented by an XG system, organizes them, and describes the behaviors.
4. The *XG Policy Language RFC* [XGPL], which describes the policy specification meta-language for implementing machine-understandable policies.
5. The *XG Policy Interface RFC* [XGPI], which describes the interface, or access methods, that can be used to pass information between an XG device and a policy reasoning engine.
6. The *XG Evaluation Platform* [XGEP], which describes the OPNET models of XG protocols.
7. The *OPNET Usage Manual* [XG-OP-USE], which describes the procedure by which the OPNET simulation system can be run.

Accomplishments under X-MAC can be classified into two broad categories – (a) the development of a framework for managing the key aspects of radio behavior through flexible application of policies, (b) design, modeling and simulation of key protocols for opportunistic spectrum access. Items (1)-(5) above fall under the framework, and the remainder under the protocol modeling and simulation. For most of the project duration, these two sets of activities were conducted in parallel.

BBN Technologies is currently performing a follow-on project to X-MAC, namely *XG Architecture and Protocols (XAP).* The work and accomplishments described here are being built upon within the XAP project.

The remainder of the document is organized as follows. We begin by presenting the background and motivation for X-MAC, and give a statement of the problem. Following that, we summarize our work under two categories – in section 3 we describe the XG framework, and in section 4 we summarize the modeling and simulation of X-MAC protocols.

2 Background

There are two significant problems confronting wireless communications with respect to spectrum use:

- *Scarcity*. The current method of allotting spectrum provides each new service with its own fixed block of spectrum. Since the amount of useable spectrum is finite, as more services are added, there will come a point at which spectrum is no longer available for allotment. We are nearing such a time, especially due to a recent dramatic increase in spectrum-based services and devices.

- *Deployment difficulty*. Currently, extensive, frequency by frequency, system by system coordination is required for each country in which these systems will be operated. As the number, size, and complexity of operations increase, the time for deployment is becoming unacceptably long.

Both problems are a consequence of the centralized, static nature of current spectrum allotment policy. This approach lacks the flexibility to aggressively exploit the possibilities for dynamic reuse of allocated spectrum over space and time, resulting in very poor utilization and *apparent* scarcity. It also mandates a priori assignment of spectrum to services before deployment, making deployment difficult.

Preliminary data indicates that large portions of allotted spectrum are unused (refer the Spectrum Policy Task Force report [SPTF]). This is true both spatially and temporally. That is, there are a number of instances of assigned spectrum that is used only in certain geographical areas, and a number of instances of assigned spectrum that is only used for brief periods of time. This wastage of assigned spectrum is bound to increase in future – spatially, due to the increasing localization of propagation due to radio devices moving up in frequency, and temporally due to the proliferation of services that are highly bursty in nature.

Studies have determined that even a straightforward reuse of such "wasted" spectrum can provide an order of magnitude improvement in available capacity. It can be concluded that the issue is not so much that spectrum is scarce, but that we do not have the technology to effectively manage access to it in a manner that would satisfy the concerns of current licensed spectrum users.

In order to address the scarcity and deployment difficulty problems, XG is pursuing an approach wherein static allotment of spectrum is complemented by the opportunistic use of unused spectrum on an instant-by-instant basis, in a manner that limits interference to primary[1] users. In other words, the basic idea is this: a device first "senses" the spectrum it wishes to use and characterizes the presence, if any, of primary users. Based on that information, and regulatory policies applicable to that spectrum, the device identifies spectrum opportunities (in frequency, time, or even code), and transmits in a manner that limits (according to policy) the level of interference perceived by primary users. We term this approach *opportunistic spectrum access.*

Opportunistic spectrum access also provides far easier deployment, or rapid entry, into regions where spectrum has not been assigned. Only minimal prior coordination is necessary, greatly easing the restrictions to meet the deconflicting requirements. This is helpful both in civilian applications such as the entry of a wireless LAN technology in less developed regions, and in military operations requiring high tempo and quick reaction time.

The realization of opportunistic spectrum access is highly challenging. Several problems must be solved: sensing over a wide frequency band; identifying the presence of primaries and characterizing available opportunities; communication among devices to coordinate use of identified opportunities; and most importantly, definition and application of interference-limiting policies, and utilization of the opportunities while adhering to such policies.

The true potential of this new approach can be exploited only if in addition to spectrum agility, we provide *policy agility* – that is, a way by which the policies controlling the behavior can be dynamically changed. That is, policies are not embedded in the radio, but can be loaded "on-the-fly". Policy agility allows adaptation to policies changing over time and geography. Further, technology (spectrum agility) can be developed in advance of policies. This is important for breaking the chicken-and-egg dilemma that exists today, where regulatory bodies must wait for technology before drafting policies and technology must wait to see what the polices will look like.

The use of policy agility using *machine readable* or *machine understandable* policies is depicted in Figure 1. Starting from the left, spectrum policies are encoded in a machine interpretable form and loaded into the XG device. The XG device then operates in accordance with its interpretation of these policies. Policies may be loaded using smart media or over the Internet. In order to change the policies we simply need to load a new version. For instance, operating in a different country would require merely downloading from a different web-site or new smart card.

Although recent years have seen some of the components for opportunistic spectrum access mature (e.g. software radios), we are a long way from a prototypical system. Further, no work exists in the area of decoupling the

[1] Users that are licensed to use the spectrum in question, subject to regulatory constraints.

policies from the implementation. This yawning gap between current state-of-the-art and what is required for opportunistic spectrum access is the motivation behind XG.

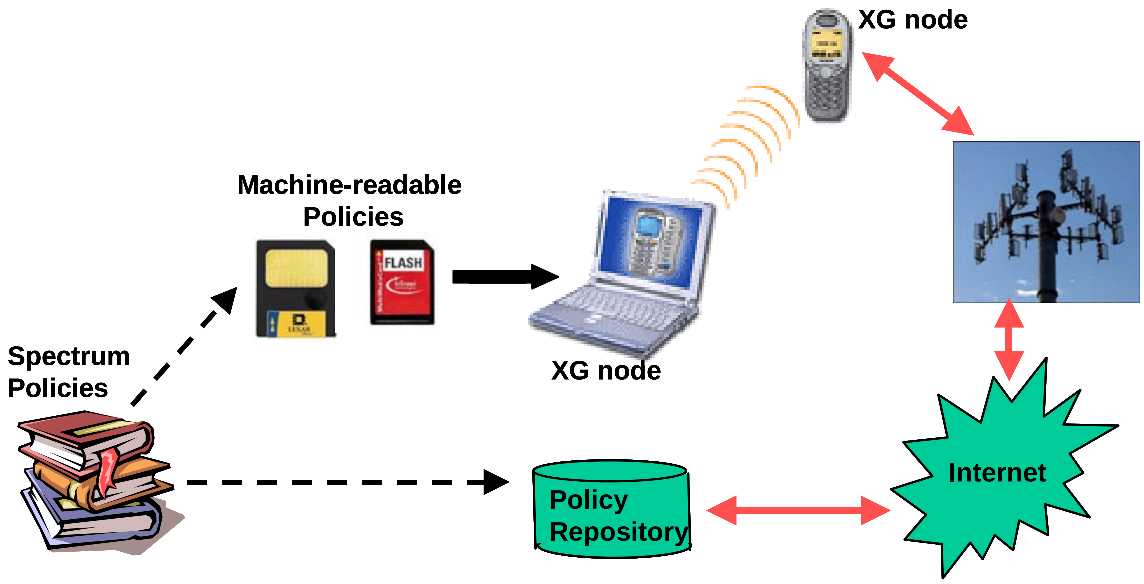

Figure 1: Machine understandable policies. With this, changing the policy merely requires loading a different flash card or downloading anew.

A more comprehensive discussion of the XG vision and motivation can be found in the XG Vision RFC [XGV].

The XG program had four performers, including BBN. The goal of the other three performers was to develop opportunistic spectrum access technologies. BBN's role within the program was unique – we served as the developer of a framework within which such technologies could reside.

Specifically, X-MAC had two main technical goals (and accomplishments):

1. *Develop a long-lived framework for managing the key aspects of radio behavior through flexible application of policies*. In order that the radio be policy-agile, we require a framework in which policies are written in a way that can be interpreted by the radio, and the radio is able to exploit such expression of policies.
2. *Develop MAC-layer protocols for XG communications and analyze them using modeling and simulation*. The development of protocols gives insight into the right framework, and modeling captures the performance variations as a function of various system and environmental parameters. Further, our simulation system serves as a "proxy XG radio" in the framework.

BBN Technologies also coordinated an XG Working Group comprising of Government and support staff, and the XG Phase I program participants.

In the remainder of this document, we summarize our work in each of these areas.

3 XG Framework

The XG framework is based on a decoupling of three fundamental elements in the design of an XG system: *policies, behaviors, protocols.* By this, we mean that policies, behaviors and protocols are defined separately with some kind of "connections" defined between them. This concept is illustrated in Figure 2.

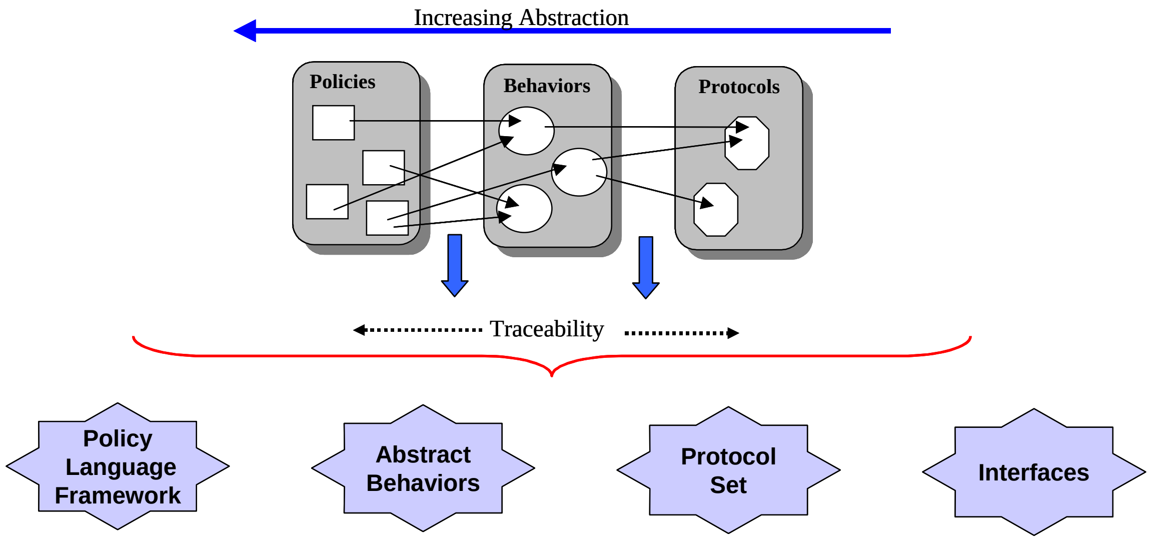

Figure 2: This illustrates the decoupling of policies, behaviors, and protocols, along with traceability and the four components of our framework.

Decoupling allows adaptation to policies that vary over time and geography. Technology can be developed in advance of policies, and worldwide deployment would be greatly simplified. Furthermore, sub-policy management, as required in secondary markets, is easier. Policies no longer have to reflect the common denominator of competing technologies, and can be tailored to the diverse system capabilities expected for opportunistic spectrum access. Decoupling behaviors from protocols allows us to control *what* needs to be done separately from *how* it is implemented, resulting in a cleaner and more flexible architecture. Indeed, we argue that a decoupling approach is not just beneficial but pretty much a requirement for harnessing the full potential of opportunistic spectrum access.

A central notion in this approach, and a notion that is enabled by this approach, is that of *traceability*. Behaviors, preferably one or more of a core set of abstract behaviors, should be traceable to policies. This provides two advantages: first, it helps make the verification of new policies easier, and second, when an XG radio is deployed in a new region, it is easier to affirm that the XG system will behave in a certain way. It allows us to accredit based on ability to correctly interpret and implement policies.

As shown in the figure, the task of decoupling policies, behaviors and protocols requires the definition of four components in the framework: policy language, abstract behaviors, protocols, and interfaces. We describe below our accomplishments on the definition of the policy language, abstract behaviors, and one interface – namely the policy language interface. Protocols were only developed as part of the parallel modeling and simulation effort within BBN (that is, unlike the language, behaviors and policy interface, this did not have program-wide scope and consequently did not involve participation of the XG working group members, namely, the Government and performers). We give an very brief overview of some of these protocols in section 4.

We reiterate that the following is only a summary. The reader should refer to the particular RFC (referenced in the respective section) for details. We also note that the development of the framework is an evolving process and this report reflects the current snapshot in that evolution. The ongoing follow on project, namely XAP, will further evolve the framework, and indeed, may modify some of the definitions given here.

3.1 Policy Language

The policy language was designed with four objectives in mind: developing a language structure that is rich enough to adequately express XG use cases, allow for machine "understandability", support inference and reasoning capabilities, and be flexible/extensible enough to be long-lived. We note that it is not the intent of our framework/language to be able to capture *all* spectrum policies, only *XG related* policies.

The key challenge that we faced is that of encoding information that has traditionally been developed for *human* consumption. Specifically, there is a vast diversity in the primitive objects that make up regulatory policy domain – from concepts of frequencies to power spectral density, mathematical formulae, geographical concepts, time concepts including zoning, possible database access etc.

To solve this problem, we have leveraged recent advances in two fields. First, the general area of *knowledge representation* has yielded tools, techniques and insight into representing human consumable information. Second, there has been considerable research into languages and tools for the *semantic web*, in particular, markup languages that encode the semantic content along with the data that are also relevant for our purposes.

In order for policies in the language to be machine readable, we need an "internal representation", that is, a standard, well-developed format for writing policy instances that is amenable to, or already has adequate tools for interpretation. Moreover, the internal representation must have the expressive power to capture the relationships in our language.

We have chosen the the DAML+OIL[DAML] language. DAML+OIL is an extension of the Extensible Markup Language (XML) and the Resource Description Framework (RDF) [RDF]. It is the combination of efforts in the US (DAML – the DARPA Agent Markup Language) and the European Union (OIL - the Ontology Inference Layer) to create a machine-readable semantic markup language. DAML+OIL provides a rich language for representing an ontology[2], that is knowledge about the interrelationship of objects, in a manner that allows a machine to make inferences.

DAML+OIL is the baseline for the World Wide Web Consortium's OWL Web Ontology Language [OWL]. We expect the XG policy language will migrate to OWL as OWL is standardized and tools mature for developing OWL content. The term "DAML" will be used throughout this report to refer to both DAML+OIL and OWL.

DAML has a unique set of features that makes it the ideal choice for XG: *inheritance and polymorphism* – to enable policy rules and properties to extend others and reduce the need for enumeration; *reification (rules about rules)*, for example, to make a policy rule governing when or where a set of policies will apply; *inference*, to infer facts that may not be explicitly stated; *extensibility* in its vocabulary, structure and semantics so that the language can adapt to express new types of policies as spectrum policy requirements change; and finally, *standards based*, so that adoption is easy and tools continue to be available.

The crux of the policy language development is to define the ontology. Before doing that, however, one needs to choose the right set of idioms for expressing policies. Using a set of example XG policies [XGPEX] as a reference point, we first converged on a set of idioms. The key insight here is that the structure of the policy language be aligned with the envisioned concept of operations in order to facilitate natural expression of the policies. In addition to description logics supported natively by DAML, the expression of XG policy requires variables, condition and constraint expressions, and implication rules. The high level idioms also help organize policy in such a way that the particular policies that apply to a type of device and its intended operating environment can be readily extracted from a potentially large collection of policies. Furthermore, this organization must allow changes to policy to be made locally without rippling over to a large set of policies, so that only a small number of policies need to be processed again. Based on these requirements, we developed the language ontology.

In sum, we accomplished the following with regard to XG policy language:

- Defined the ontology for the XG policy language, and encoded the ontology in DAML.
- Defined a "surface language" as a more human-friendly way of specifying policies.
- Developed a number of tools for creating, interpreting and reasoning with policy instances, and for conversion from the surface language to DAML
- Prepared a set of 7 policy examples as a reference point for our work

[2] An "ontology" is the knowledge about the relationships between objects. However, DAML also uses the term "ontology" to refer to a DAML file that encodes ontological information.

♦ Encoded the first example in the surface language and converted it to DAML

♦ Prepared the XG Policy Language RFC and addressed feedback from the XG working group.

♦ Led the discussion of the XG policy language and internal representation language (DAML) within the XG working group and sought consensus on key issues.

3.2 Abstract Behaviors

We developed a system model of XG within which we have identified several abstract behaviors. Our system model is shown in Figure 3. The system model shows a general high-level block diagram of an XG system, showing its key components including the transceiver, the MAC, and the XG kernel. We also show the main interfaces of the XG-kernel to the transceiver, the MAC, and the application.

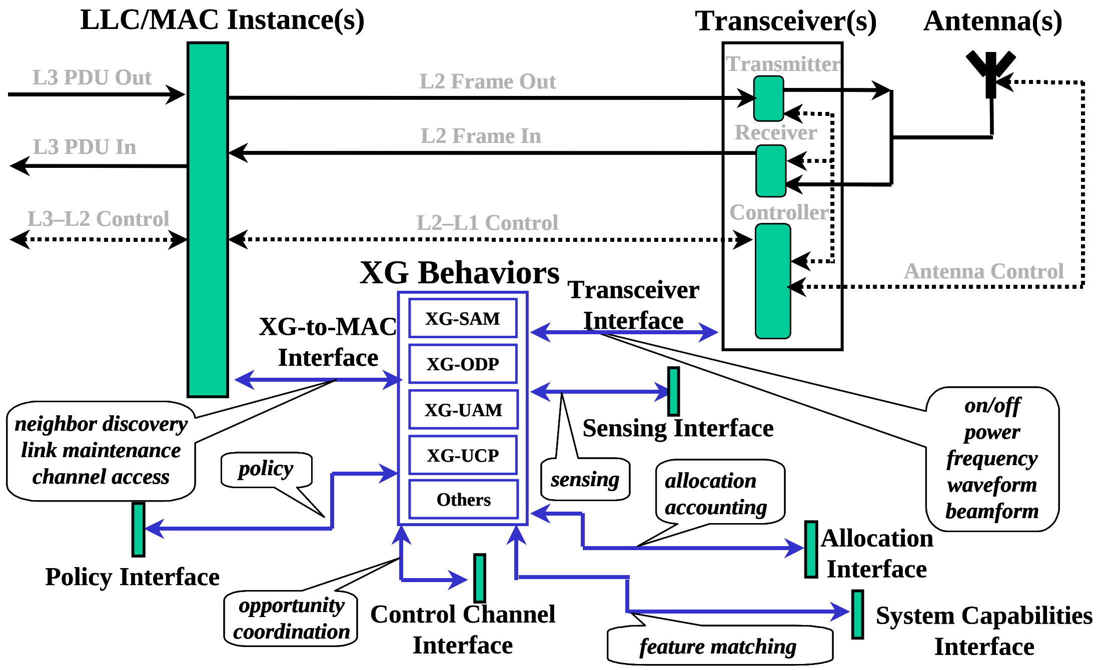

Figure 3: Notional XG system model

In this model, we show a notional XG system that has an LLC/MAC layer and an agile transceiver. The XG system implements a number of abstract *behaviors*, which interact with each other and the rest of the XG system through extensible *interfaces*. The behaviors include *protocol* behaviors that involve communications from/to an XG system as well as behaviors that are *internal* to the XG system.

We have identified the following core abstract behaviors:

1. The *XG Spectrum Awareness Management (XG-SAM)* behavior, which describes how opportunity information is acquired, identified, represented and disseminated within and across XG systems. XG-SAM encompasses awareness information gained from sensing, policy (including configuration), and through XG Opportunity Dissemination Protocol (XG-ODP) instances, as well as opportunity identification.

2. The *XG Opportunity Dissemination Protocol (XG-ODP)* behavior, which is a class of protocol behaviors that can be used by XG systems to share opportunity awareness information. An XG system should participate in one or more instances of XG-ODP classes.

3. The *XG Usage Accounting Management (XG-UAM)* behavior, which enables emissions to be traced to a valid opportunity and a set of policy rules that allows this usage. Therefore emissions must be tagged with an opportunity object and a policy object.

4. The *XG Use Coordination Protocol (XG-UCP)* behavior, which allows XG systems to coordinate the use of selected opportunities with other (XG and non-XG) systems. XG systems should participate in one or more instances of one or more XG-UCP classes.

Each of these behaviors is described in detail in the XG Abstract Behaviors RFC [XGAB]. Each behavior has a number of components, which are organized in a hierarchical, object-oriented fashion. For instance, within the XG-SAM class of behaviors, there is an awareness class based on sensing, an awareness class based on information disseminated from another XG node, etc. Sensed awareness, in turn, has sub-classes, namely location and spectrum usage.

These classes are specified using the Unified Modeling Language (UML) and easily extensible (please refer the RFC [XGAB]).

XG abstract behaviors interact with each other and with the system through extensible interface sets (refer Figure 3). We have defined the following interfaces. Further details can be found in the XG Abstract Behaviors RFC [XGAB].

1. *XG Sensing Interface:* The abstract interface through which sensed awareness (of the operational environment, i.e. information such as location and spectrum) is accessed and sensing behavior is controlled.
2. *XG System Capabilities Interface*: The abstract interface through which the capabilities and current configuration of the XG system are accessed.
3. *XG Policy Interface:* The abstract interface through which (regulatory and system) policy information is obtained.
4. *XG Control Channel Interface:* The abstract interface through which virtual control channels (which carry XG protocol data) are managed and accessed.
5. *XG Transceiver Interface:* The abstract interface through which the agile XG transceiver (i.e parameters such as transmit power, frequency, waveform, and beamform) is controlled and emission constraints are conveyed.
6. *XG Allocation Interface:* The abstract interface through which a specific opportunity is allocated for use from all available opportunities.
7. *XG-to-MAC Interface:* The abstract interface through which the MAC layer can interact with the XG abstract behaviors (for example, to support link setup and maintenance, contention management, and framing).

In sum, we have accomplished the following with regard to XG Abstract Behaviors

- ♦ Defined a notional XG system model.
- ♦ Identified four extensible abstract behavior classes and defined the inheritance hierarchy of sub-behaviors within each class.
- ♦ Identified seven extensible interface sets and defined the access methods for the XG System Capabilities interface.
- ♦ Prepared the XG Abstract Behavior RFC and addressed feedback from the XG working group.
- ♦ Led the discussion on the XG abstract behaviors within the XG working group and sought consensus on key issues.

3.3 Policy Interface

The Policy Interface is a set of access methods to extract information from the policies. In our architecture, the policy interface connects a Policy Reasoner that can understand and reason about XG policies to a Policy Agile XG Radio which executes behaviors constrained by policies. In this sense, the policy API effectively separates the XG system implementation from the task of interpreting XG policies.

Figure 4 depicts a view of the logical components involved in an XG framework. In this view, the functionality of the policy interface, which connects the XG radio and the policy database, has two main parts – *policy filter*, and *policy reasoner*. So the policy interface or API is the union of the Policy Filter API and the Policy Reasoner API. This architecture provides a great deal of flexibility. The XG radio can use access methods in the reasoner API and/or the filter API. The XG radio can also work directly with raw unprocessed XG policies (effectively by-passing the policy interface) using the pass-thru functions of the API.

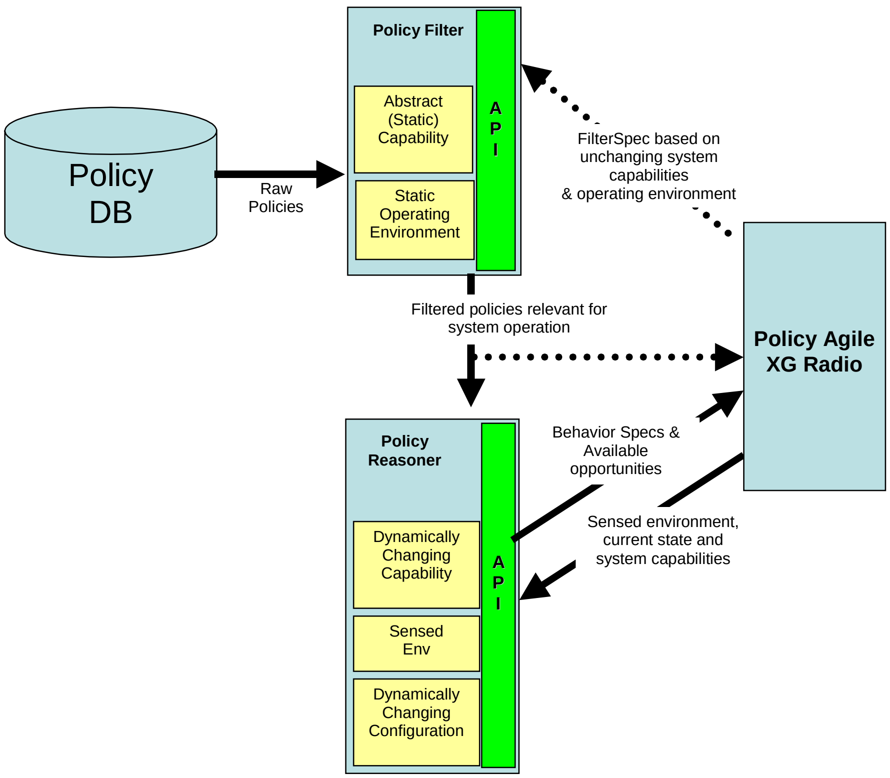

Figure 4: XG Policy Interface Architecture

The policy filter extracts relevant policies from the (potentially huge) database. Specifically, it helps identify policies applicable to a given family of XG systems or a specific XG system in its intended operating environment. The policies are filtered on "static" characteristics of the given XG system, that is, those characteristics (system or operational) that are not expected to change over the course of the XG node's operation. By identifying a subset of useful policies from the much larger database, the policy filter reduces the number of policies that an XG radio needs to carry to the field. The filter typically does not reason about policies, or deduce what opportunities might be available to the XG system – that is the job of the system policy reasoner. In the XG Policy Interface RFC [XGPI], we have defined a *policy filter interface,* or the API, which encapsulates the methods and function-calls required to access the functionality of a policy filter.

For example, an XG system that only operates in the "S" radar band, that does not have a "spectrum usage channel" and which will only be fielded in Minnesota, could use the filter API to extract the small subset of policies that are applicable to its capabilities and operating environment. This would filter out, for example, the radar policies that are not meant for the "S" band, or which require access to the "spectrum usage channel" or which are applicable only in Wisconsin.

The policy filter input, output and access methods are described in detail in the XG Policy Interface RFC [XGPI].

The system policy reasoner on the other hand can be used by the XG system as it operates in the field, to reason about policies based on the current environmental and system state. In typical operation, the policy reasoner would interpret the policies (or the small subset of device applicable policies from the filter) based on the current environmental and system state of the XG radio. Specifically, the policy reasoner will (i) identify available spectrum opportunities, (ii) identify permissible operating behaviors, or *behavior specifications*, and (iii) confirm/deny whether a specified transmission action is permitted for the XG device. In the XG Policy Interface RFC [XGPI], we will define a *policy reasoner interface*, or the API, which encapsulates the methods and function-calls required to access the functionality of a policy reasoner.

Another important functionality of the reasoner is to provide XG behavior traceability. In order to enable traceability, which is a central goal of XG (see [XGV]) the reasoner provides a function that offers a proof of the proposed opportunities or behavior. The proof is a sequence of inferences based on policy clauses which "imply" the existence of an opportunity. The proof provides traceability of emission behaviors to the set of policies that support this proof accompanying that behavior.

The policy reasoner helps the XG radio discover available opportunities, and identifies constraints on its behavior. The reasoner also acts as a policing entity by answering questions about what is permissible based on the current state and the policies. Therefore, the output of the policy reasoner can be classified into three distinct types: (i) list of opportunities, (ii) constraints on the radio's behavior (based on the known radio capabilities), and (iii) a response that answers the queries posed by the XG radio.

The policy reasoner provides two types of opportunities to the XG system: (i) Actionable and (ii) Potential. Actionable opportunities are those that are immediately available based on what the XG system is sensing, and potential opportunities are those that may become available if the XG System can provide additional information. It should be noted that potential opportunities are designed to provide a means to guide the policy reasoner to search for opportunities. In that sense, potential opportunities go beyond the task of policy interpretation, and more into the functionality of a cognitive XG radio.

The policy reasoner input, output and access methods are described in detail in the XG Policy Interface RFC [XGPI].

In sum, we have accomplished the following with regard to the policy interface

- Defined the architecture for the interface, including the entities on either side of the interface and the functional decomposition of the interface.
- Defined the access methods for the policy filter and policy reasoner parts of the policy interface.
- Given the concept of operations of using the policy interface from a radio device's perspective, based on a radar sensing example (see [XGPI] for details).
- Prepared the XG Policy Interface RFC and addressed feedback from the XG working group.
- Led the discussion on the XG policy interface within the XG working group and sought consensus on key issues.

4 Modeling and Simulation of MAC protocols for XG

We have developed key XG protocols and an OPNET model of an XG radio incorporating these protocols. The objectives of this evaluation platform are to:

- Provide a better understanding of the challenges anticipated in the design of a real life XG system. This understanding will be helpful in the Protocol and Behavior Specification that is being developed as part of the XG Request for Comments (RFC) process.
- Help validate the correctness of some of the protocols/behaviors developed under the XG's Protocol and Behavior Specification work.
- Provide insights into the range of performance gains achievable with an XG system and the complexity associated with building such a system.
- Help in the design and evaluation of notional scheduling algorithms for XG communications.

Our model partitions the functionalities of XG nodes into four layers: physical, XG, MAC, and application, as shown in Figure 5.

The *physical layer* models a frequency agile radio modem. It differs from traditional implementations in that a receiver can tune to different frequencies on-the-fly, even in the middle of receiving an interference packet.

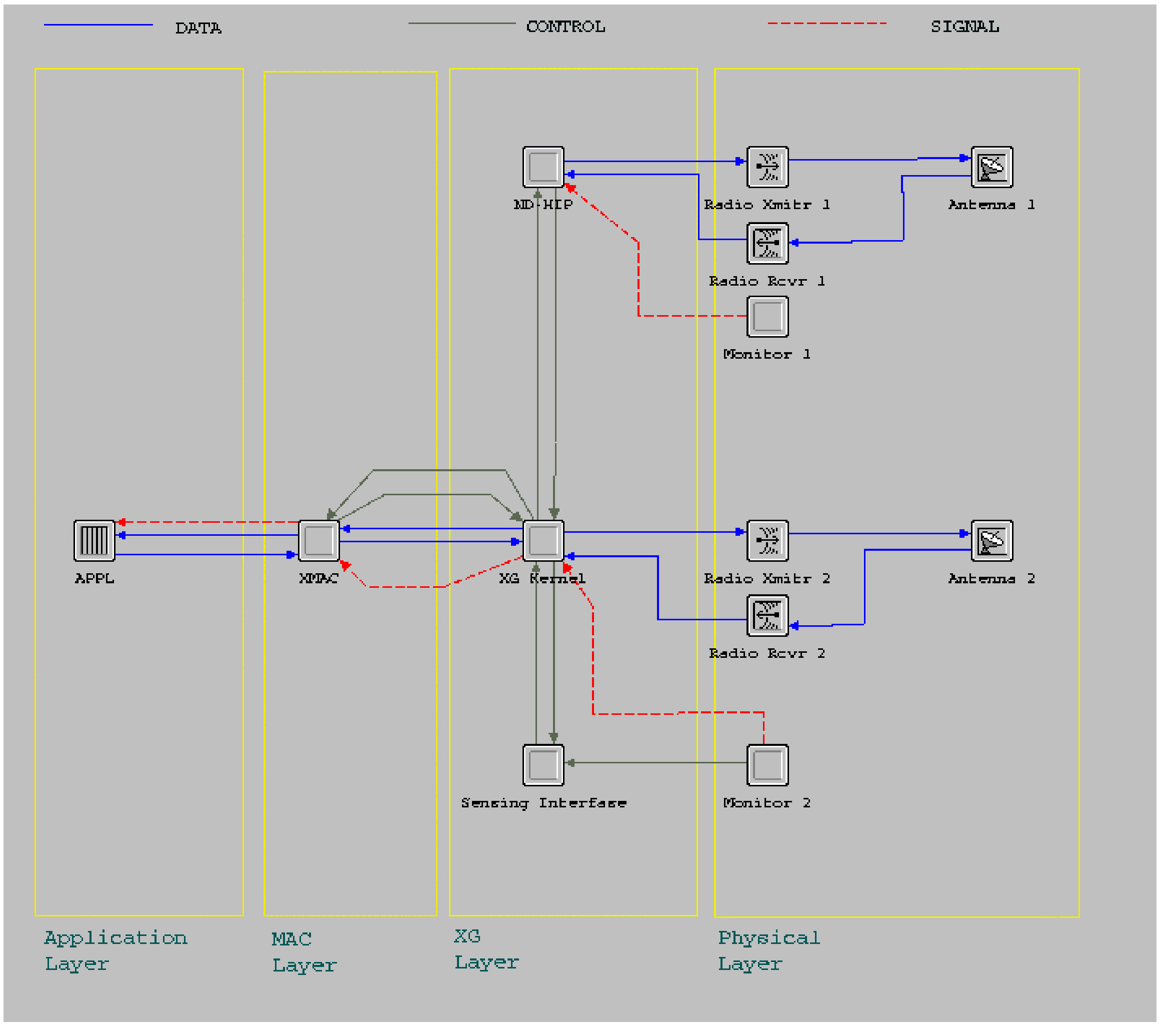

Figure 5: Secondary node model in OPNET

The *XG layer* groups the functionalities/services above the physical layer that an XG-enabled node must include in order to exploit unused spectrum even in the case that an XG-unaware MAC is used. The XG layer thus comprises the core of the XG-related functionalities. The XG layer is divided into three modules: the *XG-kernel* module, the *Sensing Interface* module, and the *Neighbor Discovery and Hole Information Protocol (ND-HIP)* module. The XG-kernel module centralizes the XG layer's decisions and presents a well-defined front-end to the MAC and physical layers. The Sensing Interface module is in charge of detecting, by means of physical sensing, the presence of primary nodes. And the ND-HIP module is in charge of disseminating critical XG information among nodes.

The *MAC layer* may be XG-aware or XG-unaware. It coordinates the access to the medium. In case of an XG-aware MAC, it allocates frequency slots to be used for transmission/reception. In this version we are implementing an XG-unaware CSMA-based MAC.

The *application layer* generates the data traffic and collects the throughput statistics. In this version, the application layer only generates and sends packets to one-hop neighbors, and therefore no routing functionality needs to be implemented.

The crux of XG – including the core XG behaviors – is in the XG layer. We summarize the XG layer protocols briefly. The details of these models, and the models in the physical, MAC and application layers are described in the XG Evaluation Platform specification [XGEP].

The XG layer comprises three modules: XG-kernel, Sensing Interface, and ND-HIP.

4.1 The XG-Kernel module.

The XG-kernel module manages the behavior of the different modules at the XG layer, selects a channel to tune while in IDLE mode – i.e. the IDLE channel, - and provides a standardized front end to the MAC layer. It receives and processes the MAC layer requests/commands using the XG Opportunity API, and coordinates the interaction between the other XG-layer modules.

4.2 Sensing Interface

The sensing interface, associated with the *XG Data Transceiver*, implements the XG Sensing Interface model. The function of the Sensing Interface module is to model the behavior of actual XG sensing hardware.

We assume the Sensing Interface module can distinguish between XG and non-XG transmissions (for example, by use of a physical layer signature), and only keeps track of primary (non-XG) node transmissions. Thus, the Sensing Interface module will ignore packets generated by secondary nodes when doing its received power computation.

4.3 ND-HIP module

The Neighbor Discovery and Hole Information Protocol (ND-HIP) module implements the neighbor discovery, pathloss computation, and hole awareness methods. The ND-HIP module broadcasts HELLO-like HIP packets so that new links can be discovered. The HIP packets will include the power used for transmission so that the pathloss can be estimated. It will include information about the set of holes available for use by this node[3]. Additionally, the HIP Packets contain information about the IDLE channel (that is, the channel that a node is tuned to when idle) used by this node, the waveform, and processing gain) that this node will use to listen to packet preambles (after a preamble is read, the reception rate and processing gain may be adjusted according to the preamble information, although that feature is not implemented in this initial version of the XG Evaluation Platform).

We have also described the *XG Transceiver API* and the *XG Opportunity API.* The XG Transceiver API is also an extension of the "Transceiver API for FCS Communications" mandatory set, documented in "the FCS Communications Mandatory API" [FCS-C-API]. The XG Transceiver API inherits all the primitives in the above set (changing the primitive name's prefix from "Tran" to "XG_Tran" so that a primitive previously called "TranXXX" is now referred to as "XG_TranXXX"), and adds two fields in the radio profile structure, and one new primitive. These additions are described in [XGEP].

The XG Opportunity API is the interface using which the MAC layer obtains opportunity information from the XG Kernel module. It is described in detail in [XGEP].

The XG model described above, and more completely in [XGEP] has been implemented in OPNET. Experimental evaluation, including performance measurements, will be done as part of the follow on XAP project.

Finally, we have designed a method by which the XG behaviors in the evalution platform can be controlled based on policy. For X-MAC the particular method chosen involves specifying the policy in a given syntax in a configuration (.ef) file for OPNET. While this is not as flexible as the policy language being developed (as described in section 3.1), it is adequate to express a number of simple policies that serves as a very preliminary proof of concept of our vision. The implementation of the design will be done as part of the follow-on XAP

[3] This information enables a node to learn about its neighbors' hole availability, so that it can select an IDLE channel reachable by all its neighbors.

contract. Additionally, the integration of the XG policy language and the OPNET evaluation platform will be addressed in the follow-on XAP contract.

The design is given in [XG-SIM-POLICY]. Here we very briefly give the main points.

Our design supports the following parameters as part of the policy configuration:

- Frequency band restrictions: Bands where XG is allowed
- Service assignment: is XG primary or secondary in band
- Signal threshold for primary detection
- Maximum transmit power in band
- Primary signal types that must be detected in band, and corresponding detection threshold
- Frequency bands where transmission is restricted
- Maximum sensor look-through period
- Minimum sensor integration time
- Maximum ON time
- Minimum OFF time

Note that this set of parameters is powerful enough to capture complex situations such as when a well-defined signal in one band instructs the XG node to vacate a different band (public safety/interruptible spectrum).

The policy set for the simulation platform consists of three different policy types:

(i) Service assignment policies

(ii) Power-constraining policies

(iii) Transmission-time constraining policies

It is assumed that the input policies are only those applicable to the current simulated radio, that is, the inference engine has already filter/translated the policies to meet the simulated-radio's capabilities. This implies that different radio capabilities can also be specified by means of different policy instances.

In sum, we have accomplished the following with regard to the XG Evaluation Platform modeling and simulation.

- Defined the protocols and OPNET models for an “XG node”, and defined the scenario for experimentation
- Implemented the protocols and models in OPNET
- Specified the design [XGEP]
- Specified the syntax [XG-SIM-POLICY] using simulated node behavior which can be controlled using simple policies

5 XG Working Group Activities

We coordinated a working group with the goal of jointly developing an XG framework. The working group is active; its members include DARPA, AFRL, their support staff, and each of the DARPA XG performers and their team members. We organized one 2-day meeting and discussed a number of issues. A *consensus item* list was formed and we achieved consensus on a number of key issues. We also organized a follow up telecon where we discussed the policy examples.

We created a listserver based mailing list for the XG working group (xg-wg@bbn.com). Email discussions on various issues facing the program were conducted.

The XG working group has a password protected web-site which contains a summary of all of the working group activities, including meeting minutes. More details can be provided by contacting the X-MAC PI (ramanath@bbn.com).

6 References

Note: All documents, except [FCS-C-API] and [SPTF], are present in the final X-MAC distribution. Any of these documents may be obtained by sending email to ramanath@bbn.com or spolit@bbn.com. We note that most of these documents are referred to as "Working Document" as they are subject to change as a result of the work in the follow- on XAP project.

[XGV] The XG Vision RFC, http://www.darpa.mil/ato/programs/XG/rfc_vision.pdf

[XGAF] The XG Architectural Framework RFC, http://www.darpa.mil/ato/programs/XG/rfc_af.pdf

[XGAB] The XG Abstract Behaviors RFC, XG WG Working Document

[XGPL] The XG Policy Language RFC, XG WG Working Document

[XGPI] The XG Policy Interface RFC, XG WG Working Document

[XGEP] The XG Evaluation Platform: OPNET Model Description, BBN Working Document

[FCS-C-API] FCS Mandatory Transceiver API, Available as part of the DARPA/ATO Future Combat Systems Communications project "UDAAN" distribution

[XG-OP-USE] OPNET Usage Manual for XG Evaluation Platform v 1.0, BBN Working Document

[SPTF] Spectrum Policy Task Force Report, FCC ET Docket no. 02-135, November 2002, http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-228542A1.pdf

[XGPEX] XG Policy Examples, XG WG Working Document

[XG-SIM-POLICY] Description of the Policy Interface to the XG evaluation platform in OPNET, BBN Working Document